WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Tuesday, January 24, 2006

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
☐	L12	L11 and (sleep or halt or sleeping or halting)	18
☐	L11	L10 and @AD<19930917	188
☐	L10	(CPU near8 ((operation or operational) near3 mode))same (mode near3 (register or control or controller))	795
☐	L9	L8 NOT l6	4
☐	L8	L7 and l3	19
☐	L7	((central processing) or CPU) same(clock adj2 generator) same (mode near3 (register or operation))	147
☐	L6	L5 and l3	15
☐	L5	((central processing) or CPU) same(clock adj2 generator) same (mode near3 operation)	128
☐	L4	((central processing) or CPU) near8 (clock adj2 generator) near8 (mode near3 operation)	3
☐	L3	l2 and @AD<19930917	30887
☐	L2	(mode adj2 (register or control)) or (contril register) or (mode control register) or (mode controller)	93388
☐	L1	(mode adj2 (register or control)) or (contril register) or (mode control register)	86891

END OF SEARCH HISTORY

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L12: Entry 12 of 18 File: USPT Oct 19, 1982

DOCUMENT-IDENTIFIER: US 4355389 A
TITLE: Microprogrammed information processing system having self-checking function

Application Filing Date (1):
19800125

Detailed Description Text (19):
In addition to the register in the logical arithmetic logic unit ALU11, the register in the CPU includes, for example, a mode register (not shown) selecting an operation mode (a normal mode, one instruction mode, and one step mode).

Detailed Description Text (23):
FIG. 6 tabulates the kind of commands and illustrates how the address field (13 bits) in FIG. 5 is used for the corresponding command. The command "Start CPU" denoted as the "0" command code instructs the CPU to start the execution of the microprogram from the address specified by the 13 bits of the address field. The command "Step CPU" which is of the command code "1" instructs the CPU to read out a microinstruction word from the address specified by the address field and to execute it. The difference between "Step CPU" and "Start CPU" is that the former stops the operation of the CPU when the CPU executes one step irrespective of the operation mode. This may be realized merely by transferring to the CPU a control signal forcing the mode register of the CPU to the one step mode when the exclusive control circuit 27 receives the command of "Step CPU".

Detailed Description Text (24):
The execution of the test program begins with setting the mode register at one instruction mode. Upon the setting, the command "Start CPU" starts a microprogram sequence and the microprogram is executed till an "END" microinstruction representing the end of the microroutine is accessed. At this point, the CPU halts. The exclusive control circuit 27 is so designed that, when the commands of "Start CPU" and "Step CPU" are executed, it waits to read out the next command from the exclusive fault test memory 21 until the execution of the commands ends and the CPU stops. When the exclusive control circuit 27 receives the "stop test" command (command code "2"), it stops reading commands from the exclusive memory 21 and shifts to a waiting state. Such a circuit design is easy for those skilled in the art and thus the details of this will be omitted.

Detailed Description Text (43):
The objective in this step is to cause the information stored in the register or the like to change depending on the presence or absence of a fault. Therefore, the microstep series to be executed here is selected from the control store 1, from such a view point. In this manner, the microprogram proceeds to the microstep S.sub.4. And as its execution is completed, the CPU in one instruction mode stops its operation. The exclusive control circuit 21 recognizes the halt of the CPU and reads out the command in the succeeding address T.sub.1,2 (FIG. 7) and executes it. The command in this address is "step CPU, S.sub.8 " so that the CPU executes the fourth microstep S.sub.8 of the instruction 3 by one step, as shown in FIG. 2. Then, the exclusive control circuit 21 further reads out the succeeding command "start CPU, S.sub.20 " and executes it. As a result, the CPU executes the microstep

series including steps S.sub.20, S.sub.21 and S.sub.22 as shown in the instruction N in FIG. 2 and then stops its operation. In this way, the respective portions in the CPU are successively driven and the results are stored in the register or the like. If necessary, further processing is continued on the information stored and the results of the processing is again stored in the register or the like. At the end of the test 1, the exclusive control circuit 21 reads out the command "start CPU, Sc" in the address T.sub.1,n and executes it.

Detailed Description Text (103):
First, the XIP signal is applied to the exclusive control circuit 27, too. The exclusive control circuit 27 so controls as to read out the command from the specified address of the exclusive memory 21 irrespective of the address of the same memory read out finally, when the XIP signal comes in during the operation checking test execution. The command operation is such that the mode register is restored to its normal mode, the CPU is started from the specified address (Sr address) of the control storage 1 and then the exclusive control circuit 27 is placed in the waiting condition, like the final four words of the operation checking test program shown in FIG. 7. In the microprogram starting from the specified address Sr, the general register is restored and the flip-flop FDIS is reset and then the operation proceeds to the input/output interruption processing routine.

Detailed Description Text (138):
To the buffer memory control circuit 232, is transferred a buffer memory inhibition signal (use inhibition mode) from a mode register (not shown). In the presence of the buffer memory prohibit signal, the buffer memory control circuit 232 does not change the inside state of the buffer memory 111, transfers a bypass signal of the buffer memory 111 to the port control circuit 231. In this manner, the port control circuit 231 operates as if no buffer memory is used. The mode register is operable in read and write by a machine language instruction to decide the operation mode of the CPU.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L9: Entry 4 of 4 File: USPT May 13, 1986

DOCUMENT-IDENTIFIER: US 4589020 A
TITLE: TV video data input apparatus

Application Filing Date (1):
19831122

Detailed Description Text (10):
FIG. 3 is a block diagram of a first embodiment of the TV video data input apparatus in accordance with the invention. In FIG. 3, a reference numeral 11 denotes a TV camera, while a numeral 12 represents an A/D converter that converts a video signal from the TV camera 11 into a digital video data. A video memory 13 for storing the digital video data utilizes a part of an internal memory of a CPU 14. A controller 15 effects such controls as changeover between buses in response to a picture sampling request from the CPU 14 and resetting upon completion of the picture sampling. A clock generator 16 forms a reference clock, and a timing generator 17 formes various timing pulses on the basis of the reference clock. A sampling rate mode register 18 is for selecting a picture sampling rate. An address generator 19 generates addresses required when a DMA data transfer is effected and has a function to store video data in the memory in the time series order thereof on the TV screen. In addition, a video signal is represented by a reference symbol VS, a reference clock by CK, a sampling clock by SCK, a composite synchronizing signal by Sync, a vertical synchronizing signal by VSE, a mode changeover signal by M, an address bus by AB, and a data bus by DB.

Detailed Description Text (11):
In operation, the video signal VS from the TV camera 11 is converted into digital video data in the A/D converter 12 driven by the sampling clock SCK from the timing generator 17. The video data are written in the video memory 13 at addresses generated by the address generator 19. During this operation, the CPU bus is placed in a holding state. In addition, before issuing a picture sampling command, the CPU 14 specifies a sampling rate mode through the sampling rate mode register 18. It is to be noted that the arrangement is such that the mode of each of the address generator 19 and the timing generator 17 is also changed over according to the contents of the register 18.

Detailed Description Text (14):
Referring now to FIG. 5 which shows a sampling clock generating section in the timing generator 17, the timing pulses A to H shown in FIG. 4 are applied to terminals represented by symbols A to H, respectively. The timing pulses A to H are fed into AND gates AND.sub.1 to AND.sub.8, which are controlled by the outputs of a mode selector 20, respectively, operated by the output M of the sampling rate mode register 18. The outputs of the respective AND gates are sent to an OR gate OR from which the sampling clock SCK is delivered. In the sampling clock generating section, the mode selector 20 is operated according to the mode specified by the CPU 14 through the sampling rate mode register 18, to apply gate signals to the AND gates AND.sub.1 to AND.sub.8 in order to select timing pulse signals to be sent to the OR gate OR. In this way, some of the timing pulses A to H are combined to form the sampling clock SCK.

Detailed Description Text (17):
Thus, it becomes possible to select four kinds of time for writing digital data in the memory by employing the thus formed four kinds of sampling clock signals SCK as conversion-starting signals for the A/D converter 12 and by sampling and holding the digital video data converted by the A/D converter 12. The selection of a sampling clock corresponding to each mode is effected by the mode changeover signal M from the sampling rate mode register 18.

Detailed Description Text (32):
Referring now to FIG. 9 which is a block diagram of the second embodiment of the invention, a reference numeral 41 denotes a TV camera 1, while a numeral 42 represents an A/D converter that converts a video signal from the TV camera 41 to a digital video data. A video memory 43 for storing the converted video data utilizes a part of an internal memory of a CPU 44. A controller 45 effects such controls as changeover between buses in response to a video data sampling request from the CPU 44 and resetting upon completion of the video data sampling. A clock generator 46 forms a reference clock. A timing generator 47 forms various timings on the basis of the reference clock. A sampling rate mode register 48 is for selecting a video data sampling rate. An address generator 49 for generating addresses required when video data are transferred in a DMA data transfer manner has means for storing video data in the time series order on the TV screen. A vertical scanning counter is denoted by a reference numeral 50, while a sampling range setting circuit 51 sets a picture sampling range. In addition, a video signal is represented by a symbol VS, a composite synchronizing signal by Sync, a vertical synchronizing signal by VSE, a mode changeover signal by M, an address bus by AB and a data bus by DB.

Detailed Description Text (33):
In operation, the video signal VS from the TV camera 41 is converted into a digital video data in the A/D converter 42 driven by the sampling clock SCK from the timing generator 47. The video data are written in the video memory 43 at addresses generated by the address generator 49. During this operation, the CPU bus is placed in a holding state, and before issuing a picture sampling command, the CPU 44 specifies a sampling rate mode through the sampling rate mode register 48. It is to be noted that the mode of each of the address generator 49 and the timing generator 47 is also changed over according to the contents of the register 48. Moreover, the CPU 44 previously instructs the sampling range setting circuit 51 to set a sampling start position and end position. The setting of the sampling start position and end position is effected by specifying, as shown in FIG. 10, coordinates (X.sub.S, Y.sub.S) of the sampling start position and coordinates (X.sub.E, Y.sub.E) of the sampling end position within a sampling range 102 in an effective scanning screen area 101.

Detailed Description Text (37):
If several kinds of sampling clock signals SCK thus formed and selectively changed over for use are employed as conversion-starting signals for the A/D converter 42 and moreover the digital video data converted by the A/D converter 42 are sampled and held, it becomes possible to selectively employ several kinds of time for writing the data in the memory. In addition, the arrangement is such that memory devices different in access time from each other can be properly changed over from one to another according to the selected write time. It is to be noted that the selection of a sampling clock corresponding to each write mode is effected by the mode changeover signal M from a sampling rate mode register 48.

CLAIMS:

1. A TV video data input apparatus comprising:

a TV camera for producing a video signal;

means for generating sampling clock signals in a plurality of modes different from each other on the basis of a reference clock;

a sampling rate mode register means for selecting a mode of said clock signals corresponding to characteristics of a TV picture, said sampling rate mode register being controlled by a CPU in a computer;

an A/D converting means for quantizing said video signal through A/D conversion on receipt of said sampling clock signals;

a memory having a short access time or a memory having a long access time provided in the computer for storing the output from said A/D converting means; and

a writing means, operatively connected with said memory and said means for generating clock signals, for writing in said memory the output from said A/D converting means,

whereby the writing of the output from said A/D converting means into said memory having a short access time at a high speed and that into said memory having a long access time at a low speed can be switched over from one to the other by selectively changing the mode of said sampling clock signals.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L12: Entry 3 of 18 File: USPT Apr 2, 1996

DOCUMENT-IDENTIFIER: US 5504908 A
TITLE: Power saving control system for computer system

Application Filing Date (1):
19930330

Brief Summary Text (6):
Conventionally, two types of functions for placing the personal computer into a stand-by state, i.e. so-called a rest mode function and a sleep mode function are provided in certain types of personal computer systems. The rest mode function is performed for automatically switching an operational clock frequency of the computer system from 16 Mhz in the normal operational mode to 1 Mhz when a CPU is held inoperative state for a predetermined period of time. If a further predetermined period is elapsed while the computer system is held in the rest mode state, the sleep mode is automatically initiated to shut down the power supply. In either mode of operation of the computer system, the normal mode operation can be resumed by operating an arbitrary key. In many cases, the predetermined period of time for initiating the stand-by mode can be arbitrary set by the user through manual setting operation.

Brief Summary Text (30):
It is further desirable that the power saving control system further comprises an inhibiting means for inhibiting operation of the control means for switching the operational mode from the normal-mode to the power saving mode when the operational state of the CPU satisfies a predetermined inhibiting condition.

Detailed Description Text (18):
Also, it is possible to control the operational mode of the CPU depending upon the temperature condition thereof for avoiding overheating. The temperature dependent control has been discussed in the co-pending U.S. Patent Application for "Drive Control System for Microprocessor", commonly owned by the owner of the present invention. The disclosure of the above-identified commonly owned co-pending U.S. Patent Application is herein incorporated by reference. In addition, it is also possible to selectively initiate the power save mode operation depending upon the periods, in which the repeated access mode is detected. Such procedure is effective for avoiding initiation of the power save mode in certain instance, such as execution of a loop for a software time. The process is disclosed in the co-pending U.S. Patent Application for "Power Saving Control System for Computer System with Feature of Selective Initiation of Power Saving Control" commonly owned by the owner of the present invention. The disclosure of the above-identified commonly owned co-pending U.S. Patent Application is also herein incorporated by reference.

CLAIMS:

12. A power saving control system as claimed in claim 11, which further comprises an inhibiting means for inhibiting operation of said control means for switching the operational mode from said normal mode to said power saving mode when the operational state of said CPU satisfies a predetermined inhibiting condition.

Previous Doc Next Doc Go to Doc#

Previous Doc Next Doc Go to Doc#
First Hit Fwd Refs

Generate Collection

L12: Entry 6 of 18 File: USPT Apr 26, 1994

DOCUMENT-IDENTIFIER: US 5307464 A
** See image for Certificate of Correction **
TITLE: Microprocessor and method for setting up its peripheral functions

Application Filing Date (1):
19901203

Brief Summary Text (48):
An address register selection order or task execution order specified by a programming language is converted by the program development apparatus described above into displayed data representing a selection order to be input to a means for storing a plurality of address register selection orders. In addition, the task registers are each associated with different task numbers. It thus becomes possible to specify a task execution order by software without taking the hardware configuration into consideration. Accordingly, the invention allows anybody to specify a task execution order easily, requiring no knowledge of the hardware. Moreover, the execution order of tasks can be switched with ease continuously without halting the operation of the apparatus.

Brief Summary Text (55):
In addition, according to a means more concrete than the one described above, memory addresses at which execution programs of a task are stored can be sequentially generated by using the contents of an address register associated with one of the tasks as a base. In the case an execution program shared among two or more tasks, however, it becomes necessary to identify which task is currently using the execution program. Therefore, a means is provided for generating memory addresses which are different from task to task using a task identification code. Such addresses can be generated by allocating memory areas to execution programs of a task which do not overlap those allocated to execution programs of other tasks. As another alternative such addresses can also be generated by locating only the start address of the execution program of a task or only part of the execution program including the start address at a memory area which does not overlap that allocated to another task. According to a means even more concrete than the one described above, the sequence of the execution program in each task can also be changed dynamically. To be more specific, the execution program of a task can be controlled so as to carry out initialization, halt the execution temporarily or jump to an arbitrary address at any time. Therefore, this means can dynamically control not only the sequence of tasks but also the sequence of the execution program in each task. Even when a plurality of tasks are executing concurrently, the dynamic sequence control can be performed at the execution program level.

Brief Summary Text (56):
According to a fourth configuration of the invention, a microprocessor having a peripheral function implementing means for executing a plurality of tasks on a time-division basis one after another is provided with specific information referred to as task null information which denotes that none of the tasks are to be executed. When the task null information is specified, the peripheral function implementing means enters an idle state. The idle state can be attained by halting the function to execute a task for a predetermined period of time.

Brief Summary Text (57):
A typical means according to the fourth configuration is described in more concrete terms as follows. The idle state can be attained by halting the operation to read a memory unit for storing execution programs of tasks for a predetermined period of time. As another alternative, the output of the memory unit is converted to a fixed value disabling the operation of a functional circuit for executing a program regardless of data read from the memory unit.

Detailed Description Text (5):
Based on the combination of values of signals provided through external pins MD0 to MD2, the mode control circuit 102 determines the operation mode of the CPU 2 which is output as a mode signal (MS). In response to the mode signal (MS), the CPU 2 enters either single chip mode or externally extended mode. In single chip mode, the microprocessor 1 operates as a complete microcomputer system with the CPU 2 being able to access only an address space comprising the RAM unit 3, the ROM unit 4 and on-chip registers embedded in the microprocessor 1. In externally extended mode, on the other hand, the CPU 2 can also access typical ROM and RAM units connected externally to the microprocessor 1 in addition to the embedded RAM unit 3, the embedded ROM unit 4 and the on-chip registers. That is to say, the address space accessible by the CPU 2 operating in external extended mode is extended by off-chip RAM and ROM units. In externally extended mode, the microprocessor 1 shown in FIG. 1 therefore constitutes an extended configuration of the microcomputer system in conjunction with external memory units. Externally extended mode can be minimum mode which supports an address space of 64K bytes or maximum mode supporting an address space of up to 1M bytes.

Detailed Description Text (7):
The mode signal EPM is supplied to the CPU 2, ROM unit 4, sub-processor 5 and I/O port 6. Receiving the mode signal EPM, the CPU 2 enters a halt state, the ROM unit 4 and the EPROM units of the sub-processor 5 enter write mode and the input/output port 6 enters a state able to execute functions writing code into the EPROM units. Viewed externally, in this state the microprocessor 1 appears merely as an EPROM unit.

Detailed Description Text (180):
Every bit stored in the second control memory unit 150 denotes one of the microaddress registers (NAR) 140 which are each associated with a task. Accordingly, each bit denotes a task. A task selection in a program is translated by the program development apparatus into a row of bits shown in FIG. 23. Each row comprises bits which, from left to right, correspond to the tasks T1 to T5. A bit set to a `1` selects its corresponding task and a reset bit indicates that its task is not selected. Accordingly, a row having all the bits reset to zeros corresponds to the NOP specification. If a task is to be selected by the external signal 171, the row specifying the task selection includes a plurality of bits with a value of `1`. The means (AP) 160 for selecting one of a plurality of programs (or a plurality of address register selection orders) is further equipped with a means for switching a range in which an address pointer to the second control memory unit 150 can be varied. The range is switched from one to another by this means in response to the external signal 161. The four programs stored in the second control memory unit 150 can thus be selected freely. That is to say, the task execution order can be switched from one to another continuously without halting the operation of the sub-processor 5.

Detailed Description Text (251):
As described previously, when the microprocessor 1 is reset, all the registers in the file 201 are cleared. However, instead of clearing the registers, the task identification number read register 102 shown in FIG. 39 can be forcibly loaded with the task null code in order to set all the pipeline stages after TRD into a halt state. In this way, the pipeline control can prevent a malfunction from

occurring after the microprocessor exits the reset state, resuming its operation correctly.

Detailed Description Text (252):
Even during a normal operation, the task identification number read register 102 shown in FIG. 39 can be forced to output the task null data as well in order to temporarily halt the execution of tasks at any time for an arbitrary period. In this way, the pipeline execution can be put in a functionally halted state continuously without destroying information remaining at TDC and the subsequent stages. That is to say, the sub-processor 5 can be put in a temporary halt state while retaining its internal status. As the sub-processor 5 exits the temporary halt state, the pipeline execution also undergoes a transition from the temporarily halted state to a normally operational state in an orderly manner, resuming the operations based on the task information left in each pipeline stage at the time the sub-processor 5 entered the temporary halted state. The operation is thus restarted like the resumption following a reset state.

Detailed Description Text (265):
In addition, in an execution pipeline scheme, pipeline initialization and a temporary halt function can be implemented by operations on an early pipeline stage only. To be more specific, by merely forcing a task execution sequence output circuit to output task null data, the pipeline initialization and temporarily halt function can be implemented. Therefore, no circuit is required for providing information for the pipeline initialization and temporary halt function to subsequent pipeline stages at timings delayed appropriately for each stage.

Previous Doc Next Doc Go to Doc#